

PROM: Personal Knowledge Graph Construction with Large Language Models

Aishan Maolinyazi¹^a, Chaohong Ma²^b, Xiaofeng Meng¹^c and Bingbing Xu¹^d

¹Renmin University of China, Beijing, China

²HEBEI Normal University, Shijiazhuang, China

Keywords: Personal Knowledge, Knowledge Graph Construction, Large Language Models.

Abstract: The growing volume of digital information requires effective Personal Knowledge Management. Personal Knowledge Graphs (PKGs), which model knowledge as connected entities and relationships, show potential. Chats or natural voice conversations contain abundant context information about users' thoughts and preferences, which is beneficial for constructing PKGs. However, constructing PKGs from unstructured natural conversations is still challenging. The main obstacle comes from two aspects: inherently complex and context-dependent conversations. In this paper, we present **PROM**, a novel framework of **personal knowledge graph construction with LLMs**. PROM effectively constructs PKGs from natural conversations. Particularly, PROM constructs PKGs with rich knowledge information, preserves context information for knowledge provenance, and fuses different kinds of contexts for structural and semantic coherence. Specifically, PROM constructs knowledge triples (subject, predicate, object) from conversational text and integrates them into a coherent PKG with the help of LLMs. We propose a multi-strategy knowledge fusion technique to resolve conflicts and unify information from different sources for structural and semantic consistency. Moreover, we design an API proxy engine to facilitate consistent knowledge extraction from different LLM backends. The proxy system is flexible and cost-effective. It can adapt different triple extraction strategies from LLMs and unify the results with a knowledge fusion strategy. We evaluate PROM in different scenarios. The experiments show that PROM is able to construct comprehensive and context-aware PKGs from unstructured conversations and can support personal knowledge discovery.

1 INTRODUCTION

The proliferation of digital information has made effective Personal Knowledge Management (PKM) systems highly desirable (Apshvalka and Wendorff, 2005). Individuals are constantly confronted with streams of information from diverse sources, including emails, messages, documents, and social networks. Much of this valuable information originates from naturally occurring conversational data, such as transcripts and voice calls (Fu et al., 2020). To cope with this data deluge, there is a critical need for effective techniques to automatically manage, distill, and access relevant knowledge tailored to both personal needs and specific contexts. Personal Knowl-

edge Graphs (PKGs), which model entities as nodes and relations as edges, offer a particularly promising approach for this task, enabling efficient knowledge discovery and personalized insights.

Traditional methods for knowledge graph construction, typically reliant on rule-based or statistical machine learning approaches (Paulheim, 2017), often demand significant manual labor for feature labeling and rule design. This inherent limitation hinders their scalability across various personal knowledge domains and contexts. Furthermore, these methods prove less effective when confronted with the highly ambiguous and context-dependent natural language found in conversations (Liu et al., 2020), rendering them less applicable for dynamic, personalized scenarios. Traditional PKG methods rely on manual templates/ontologies, which makes it difficult to handle unstructured conversations and lack privacy awareness mechanisms.

The recent advent of Large Language Models

^a <https://orcid.org/0009-0006-1376-6701>

^b <https://orcid.org/0009-0001-6168-4771>

^c <https://orcid.org/0000-0002-7889-2120>

^d <https://orcid.org/0009-0006-5253-7338>

(LLMs) has opened new avenues for automated knowledge graph construction (Yao et al., 2019). Pre-trained on massive text corpora, LLMs demonstrate remarkable capabilities in understanding natural language nuances, resolving ambiguities, and performing complex reasoning tasks (OpenAI, 2023). These strengths make them exceptionally well-suited for extracting structured knowledge from unstructured conversational data and constructing coherent PKGs. While several studies have explored LLMs for knowledge graph completion and entity linking (Cheng et al., 2023), constructing and maintaining PKGs specifically from conversational data using LLMs poses unique challenges. These include preserving complex conversational context, effectively fusing knowledge from heterogeneous sources, and adapting to users' evolving knowledge preferences. Effectively addressing these challenges is essential to fully harnessing the potential of LLMs in personal knowledge management.

To address these challenges, this paper proposes **PROM**, a novel framework for personal knowledge graph construction with LLMs. PROM efficiently constructs PKGs directly from natural conversations, specifically emphasizing context preservation and seamless knowledge fusion to enhance personal knowledge management.

Our framework incorporates several key designs: First, an **API Proxy Engine** facilitates robust interaction with LLMs, preserving the utility of the original API and enabling consistent access to conversational data. Second, a **Knowledge Extraction** module leverages the few-shot learning capabilities of LLMs to extract structured knowledge in the form of triples (subject, predicate, object) from unstructured conversational text. This approach minimizes reliance on extensive fine-tuning and adapts to diverse conversational styles. Third, **Knowledge Fusion** implements a multi-strategy approach to resolve conflicting information and consolidate knowledge from heterogeneous conversational sources, utilizing semantic similarity metrics and rule-based conflict resolution to ensure consistency and completeness.

Building on these designs, we develop a comprehensive framework that leverages prompt-engineered LLMs and graph databases to enable efficient PKG construction. Comprehensive experiments demonstrate PROM's superiority in real-world personal knowledge management scenarios, highlighting its effectiveness, adaptability, and generalizability.

The main contributions of this work are summarized as follows:

- We propose **PROM**, a novel framework for constructing personal knowledge graphs from conver-

sational data, leveraging the power of LLMs to achieve scalability, adaptability, and minimal fine-tuning.

- We develop an **API Proxy Engine** that enables consistent knowledge extraction across different LLM backends, facilitating future expansion to include new and improved LLMs as they become available.
- We design a **knowledge extraction module** that employs prompt engineering with few-shot learning to increase extraction accuracy and reduce reliance on labeled data for triple extraction.
- We introduce a **multi-strategy knowledge fusion mechanism** that enhances graph connectivity and coherence by intelligently integrating information from diverse sources and conversation types, resolving conflicts using heuristics, confidence scores, and optional user input.

The remainder of this paper is structured as follows: Section 2 reviews the related work. Section 3 details our PROM framework. Section 4 presents the experimental setup and evaluation methodology. The experimental results and discussion are provided in Section 5. We conclude the paper in Section 6.

2 RELATED WORK

Personal Knowledge Graph (PKG) construction is a burgeoning area focused on organizing an individual's unique knowledge, preferences, and experiences to facilitate personalized information management. Extensive research highlights the diverse applications of PKG across fields like health, finance, education, and general research (Chakraborty and Sanyal, 2023). These efforts often detail various construction methodologies, including those leveraging user-generated content, system-stored data, and hybrid approaches. The construction of PKG in this area leverages both traditional and deep learning-based strategies, spanning multiple domains, such as education, business, and healthcare. We next discuss the PKGs' construction approaches from traditional and deep learning-based aspects.

Traditional PKG Construction Methods. Early approaches often relied on rule-based or crowdsourced systems. For example, the teaching system in (Weng et al., 2020) used crowdsourcing and reverse captcha to build hierarchical graphs with user collaboration. In minority languages, (Sun and Zhu, 2016) used SVM and template-based extraction for Tibetan knowledge, supporting question answering.

In domain-specific applications, (Yu et al., 2020) proposed the Tax Knowledge Graph for modeling U.S. and Canadian tax rules. (Vassiliou et al., 2024) introduced iSummary, which creates personalized graph summaries by analyzing query logs.

Deep Learning-Based PKG Construction Methods. Several methods leverage advanced deep-learning architectures to construct PKG. For example, (Jiang et al., 2019) introduced AttKGCN, which uses attribute-aware GCNs for person re-identification. In healthcare, (Shirai et al., 2021) reviewed challenges in Personal Health Knowledge Graphs (PHKGs), while (Seneviratne et al., 2021) proposed a semantic dietary knowledge model for personalized recommendations.

Recent works explore LLMs for graph construction. (Çöplü et al., 2024) leveraged LLMs with ontologies to capture personal information. (Yang et al., 2021) proposed TrigNet, a tripartite graph using psycholinguistic features for personality detection. (Kuculo, 2023) focused on event-centric knowledge graphs using transformer-based extraction.

Although these efforts are valuable, most do not fully exploit LLMs for dynamic, context-aware PKG construction from natural conversations. This work addresses challenges by proposing a framework for PKGs with strong context preservation and knowledge fusion capabilities.

3 THE FRAMEWORK

In this section, we first present an overview of PROM and then detail each component.

3.1 PROM Overview

As depicted in Fig. 1, PROM consists of three main components: the API Proxy Engine, Knowledge Extraction, and Knowledge Fusion. The **API Proxy Engine** uses a Forward API to retrieve and collect conversation data from the user applications (e.g., ChatBox, Zotero). It supports batch and streaming data from various sources, including text documents, APIs, and direct user input. Next, the **Knowledge Extraction** identifies and extracts conception, entity, relation, and triplet using a hybrid and fallback approach that combines LLM-based methods with rule-based techniques. Subsequently, the **Knowledge Fusion** module refines the extracted knowledge through entity resolution, merging strategies, relationship discovery, and triplet inference. This module also systematically addresses inconsistencies, removes duplicates, and completes missing

relations through knowledge correction to ensure high-quality PKGs. It also provides insight through entity, relation, and triplet analysis. Finally, the resulting PKGs are stored in the Knowledge Storage.

3.2 API Proxy Engine

Personal data is distributed across various applications in different formats, making efficient collection a challenge. PROM needs to automatically and efficiently collect personal conversation data from these applications or clients. However, native LLM APIs are often insufficient for direct conversational data extraction. In addition, different applications often expose diverse interfaces, necessitating tailored handling.

To address these challenges, the API Proxy Engine serves as a unified foundation, integrating multiple Large Language Models (LLMs), such as Llama, Qwen, and Deepseek. This layer enables automated and efficient collection of conversation data from a variety of sources while abstracting the differences between underlying LLMs. Specifically, the Forward API component replicates the original LLM APIs and retains their utility functions, forwarding conversation data to the Knowledge Extraction module for further processing.

3.3 Knowledge Extraction

The knowledge extraction process aims to leverage the power of LLMs while incorporating safeguards to ensure accuracy and consistency. This is achieved through a fallback mechanism: we define a confidence threshold to determine whether the LLM has effectively extracted the required knowledge. If the confidence score produced by the LLM falls below this threshold, the system reverts to a rule-based extraction method. Next, we describe the LLM-based extraction and the associated fallback mechanism.

LLM-Based Extraction: We use prompt engineering to guide the LLM (specifically, GPT-4(OpenAI, 2022)) in extracting subject-predicate-object triples (i.e., knowledge) from various data sources. The prompts are designed to elicit structured output and include instructions for providing confidence scores for each extracted triple. An example *prompt* is: “Extract all subject-predicate-object triples from the following text and provide a confidence score (0-1) for each triple: [text]”. The following function can represent the process:

$$(S_i, P_i, O_i, c_i) = \arg \max_{s,p,o} LLM(Text, Prompt) \quad (1)$$

Where *Text* is the input text, *Prompt* is the crafted

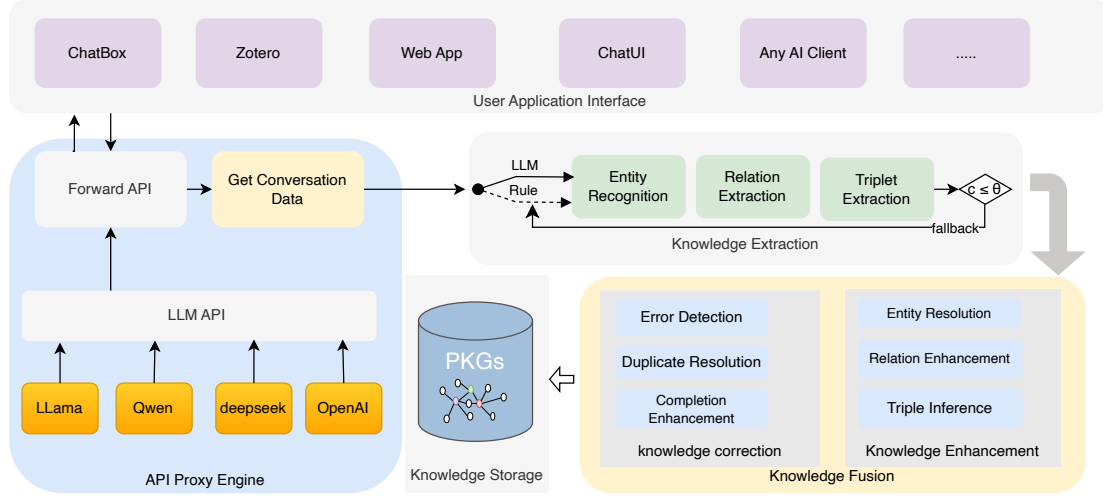


Figure 1: The architecture of PROM.

LLM static prompt, $LLM(Text, Prompt)$ represents the LLM's output given the text and prompt, S_i, P_i, O_i represents the subject, predicate, and object of the i -th triple, c_i is the LLM's confidence score for the i -th extracted triple, ranging from 0 to 1.

Fallback Mechanism: To address potential LLM errors or low-confidence extractions, we implement a rule-based fallback mechanism using regular expressions and predefined patterns. This ensures a baseline level of knowledge extraction even when the LLM performs poorly. In the implementation, a confidence threshold is defined to determine whether an extracted triple should be retained for knowledge graph construction or if the system should fall back to rule-based extraction, as described in Algorithm 1.

Algorithm 1: Triple Extraction with Confidence Score.

Data: Input Text, LLM Model, Prompt, Rule-Based Extraction $Regex$, Confidence Threshold θ

Result: Set of Triples with Confidence Scores $\{(S_i, P_i, O_i, c_i)\}$

$(S_i, P_i, O_i, c_i) \leftarrow LLM(Text, Prompt) \triangleright$ Apply LLM to extract triples and confidence score;

if $c_i < \theta$ **then**

$\{(S_j, P_j, O_j, c_j)\} \leftarrow Regex(Text) \triangleright$ Apply rule-based extraction, default confidence score $c_j = \theta$;

return $\{(S_j, P_j, O_j, c_j)\}$;

else

return $\{(S_i, P_i, O_i, c_i)\}$;

Let $Regex(Text)$ be the rule-based extraction

function; if the LLM provides a confidence score below a certain threshold θ , it uses the regular expression for information extraction approach (Li et al., 2008).

$$Triples = \begin{cases} LLMExtract(Text, Prompt), & \text{if } c > \theta \\ Regex(Text) & \text{if } c \leq \theta \end{cases} \quad (2)$$

In summary, the knowledge extraction process powered by LLMs preserves the original text and source information alongside each extracted triple. This approach maintains contextual integrity and ensures provenance, thereby facilitating auditing. As a result, users can easily trace the origin of any piece of knowledge within their PKG.

3.4 Knowledge Fusion

After extracting the triplets, further refinement is needed to improve the completeness and quality of the knowledge graph. We thus design Knowledge Fusion to systematically transform extracted knowledge into a coherent, high-quality PKG. Specifically, the Fusion process is a two-stage pipeline, containing Knowledge Enhancement and Knowledge Correction, which ensures both structural consistency and semantic completeness of the resulting knowledge graph.

Formally, let $G = (E, R)$ represent a knowledge graph where $E = \{e_1, e_2, \dots, e_n\}$ is the set of entities, $R = \{r_1, r_2, \dots, r_m\}$ is the set of relations, both extracted through the above Knowledge Extraction process (Section 3.3). Together, E and R form a set of triples $T = (e_s, r, e_o)$, where e_s, e_o are the head and tail entities separately and r denotes their corresponding relation. Each triple $t = (e_s, r, e_o) \in T$ has an as-

sociated confidence score $\omega(t) \in [0, 1]$, indicating extraction reliability (as described in Section 3.3). The fusion process aims to transform an initial graph G_0 into a refined graph G^* :

$$G^* = C(\mathcal{A}(G_0)) \quad (3)$$

where $\mathcal{A}$ and C denote the Enhancement and Correction operators, respectively.

3.4.1 Knowledge Enhancement

The Enhancement process operates in two sequential steps: graph analysis followed by knowledge integration.

Step 1: Graph Analysis. We analyze the current knowledge graph G_0 to understand its structure and quality. This analysis produces a profile $P = (S(G_0), C(G_0), A(G_0))$ where $S(G_0)$ represents structural metrics, $C(G_0)$ denotes the completeness metrics, and $A(G_0)$ represents the average confidence score. Specifically, we compute $S(G_0) = \frac{|E_{connected}|}{|E|}$ as the connectivity ratio where $E_{connected}$ denotes entities in the largest connected component. We measure completeness as $C(G_0) = 1 - \frac{|E_{isolated}| + |R_{missing}|}{|E| + |R_{expected}|}$, where $E_{isolated}$ are disconnected entities and $R_{missing}$ are expected but missing relations. We calculate $A(G_0) = \frac{1}{|T|} \sum_{t \in T} \omega(t)$ as the average confidence score across all triples. The profile components directly map to the overall quality assessment:

$$Q(G) = \alpha \cdot S(G_0) + \beta \cdot C(G_0) + \gamma \cdot A(G_0) \quad (4)$$

where α, β, γ are weighting parameters, and $P = (S(G_0), C(G_0), A(G_0))$ provides the concrete measurements for quality evaluation.

Step 2: Knowledge Integration. Using the analysis profile P and quality score Q from Step 1, we enhance the graph through three specific operations:

Entity Resolution: We identify and merge duplicate entities using two strategies: (1) name-based matching using Jaro-Winkler string similarity with a threshold(0.85), and (2) type-based clustering where entities with identical types and high semantic similarity are grouped together. When duplicates are found, we keep the entity with the highest confidence score and merge its properties.

Relation Enhancement: We standardize relation names by mapping different surface forms to canonical representations. For example, “works_at”, “employed_by”, and “is_employee_of” are all mapped to a single canonical relation “employed_by” using embedding-based similarity matching.

Triple Inference: The new triples are generated through two approaches: (1) structural rules (e.g., if A works_at B and B is located_in C, then infer A

works_in C), and (2) LLM-based inference, where we prompt the LLM to suggest missing relationships based on existing triples:

$$T_{new} = T_{structural} \cup T_{LLM} \quad (5)$$

where $\mathcal{R}_{struct}$ represents rule-based structural inference (transitive and symmetric completion), $T_{context}$ provides contextual triples for LLM-based generation, and Θ_{prompt} represents the prompting strategy. Confidence thresholds filter extracted triples to ensure quality.

3.4.2 Knowledge Correction

The Correction operator C systematically identifies and resolves inconsistencies, duplicates, and missing relations to ensure PKG quality. The correction process operates through three sequential steps with clear relationships between intermediate results. We identify structural inconsistencies, including transitive path violations, semantic conflicts between contradictory relations, and redundant connections:

$$T_{errors} = T_{conflicts} \cup T_{inconsistent} \cup T_{redundant} \quad (6)$$

We identify duplicate triples and select the highest-confidence instances:

$$T_{unique} = \{\arg \max_{t \in T} \omega(t) : T \in \text{Duplicated}(T)\} \quad (7)$$

Based on the analysis from Steps 1 and 2, we define removal and addition sets. The removal set T_{remove} combines detected errors and duplicate instances:

The final corrected graph, constructed from triples, is obtained by:

$$T_{final} = (T \setminus T_{errors} \cup (T \setminus T_{unique})) \quad (8)$$

The correction process operates iteratively until convergence, ensuring the final PKG maintains both consistency and completeness.

4 EXPERIMENTS

This section details the experimental setup and evaluation methodology used to assess PROM’s performance.

4.1 Experimental Setup

In this section, we present the experimental setup, including the implementation details and the datasets.

4.1.1 Implementation Details

The PKG construction system was implemented using the following technologies:

Languages and Libraries: The entire system was developed primarily in Python 3.11. Key libraries and frameworks include Django 4.x for the backend, SQLite 3.9 for relational data, and Neo4j 5.20 for graph data storage.

Backend Framework: Django 4.x with Django REST framework was chosen for its rapid development capabilities and robust ORM.

Database: SQLite 3.9 was used for relational data storage. The knowledge graph data was efficiently stored and retrieved using a dedicated Neo4j 5.20 database, leveraging its optimized indexing strategies for fast query processing of triples.

LLM Integration: The proxy engine is designed to interface with both OpenAI and Claude models, facilitating dynamic selection based on availability and performance. Notably, the Large Language Models themselves were accessed directly via their REST APIs without any specific fine-tuning or modification within our system.

Evaluation Framework: A suite of custom Python-based evaluation modules was developed for assessing various aspects of the PKG. The proxy engine was configured to handle request routing and load balancing across different LLM providers. The evaluation framework was designed to automate the evaluation process, providing comprehensive metrics for assessing the quality and utility of the PKG.

4.1.2 Dataset

The experimental evaluation leverages a conversation dataset designed to simulate real-world user interactions with LLM APIs via our proxy engine. The dataset's characteristics are as follows:

User Background: Ph.D. students engaged in academic research discussions, posing questions, and seeking solutions. **Data Volume:** 40 independent conversations. **Content Scope:** Conversations spanned academic research topics, including technical discussions, knowledge sharing, and problem-solving. **Conversation Structure:** Each conversation contained approximately 50 messages. **Message Length:** Individual messages typically ranged from 100 to 500 words. **Data Modality:** Only the textual content of the dialogue messages was utilized. **Origin:** Sourced from real user interactions within our proxy engine, providing an authentic representation of LLM engagement within an academic context.

4.2 Evaluation Methodology

We employ a comprehensive, multi-dimensional evaluation framework to assess the quality and utility of the constructed PKG from various perspectives. The evaluation covers knowledge graph completeness, domain coverage, embedding-based evaluation, link prediction performance, RAG-based utility evaluation, and the impact of knowledge graph correction. Below, we detail each evaluation aspect and its corresponding metrics.

4.2.1 Completeness Evaluation Metrics

We assess completeness, accuracy, consistency, and structural quality of the knowledge graph using traditional metrics:

Entity Count and Type Distribution: $|V|$ represents the number of entities in the knowledge graph, providing a measure of its breadth. $P(t)$ represents the distribution of entity types t across the set of all entity types T , indicating the diversity of concepts captured in the KG. We can express the average number of entities per type as: $\frac{|V|}{|T|}$.

Relation Diversity: $|R|$ represents the number of distinct relations in the knowledge graph, indicating the variety of relationships captured between entities.

Triple Confidence Distribution: $P(c)$ represents the distribution of confidence scores c across all triples, where c ranges from 0 to 1. This provides insights into the reliability of the extracted knowledge.

Graph Density and Connectivity: $\delta(G)$ represents the graph density, a measure of how connected the entities are in the graph, and $\alpha(G)$ represents the average node degree, indicating the average number of connections each entity has. Graph density can be calculated as: $\delta(G) = \frac{2|E|}{|V|(|V|-1)}$, where $|E|$ is the number of edges (triples) in the graph.

4.2.2 Domain Coverage Evaluation Metrics

We evaluate the PKG's ability to represent domain-specific knowledge through two complementary metrics:

Domain Vocabulary Coverage: $Cov(V, D) = \frac{\sum_{t \in V \cap D} importance(t)}{\sum_{t \in D} importance(t)}$, which measures the proportion of important domain terms covered by the KG. Here, V is the set of PKG entities, D is the domain vocabulary, and $importance(t)$ is the importance weight of term t .

Contextual Completeness: $CC(G, \{(e_i, e_j)\}) = \frac{|\{(e_i, e_j) \in G\}|}{|\{(e_i, e_j)\}|}$, which measures the extent to which semantically related entities are connected in the KG. Here, G is the PKG and $\{(e_i, e_j)\}$ is the set of expected entity pairs.

4.2.3 Embedding-Based Evaluation Metrics

We use entity embedding techniques ((Grover and Leskovec, 2016)) to evaluate the semantic quality of the knowledge graph. This involves analyzing entity clustering and measuring classification accuracy based on entity embeddings. Specifically, Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI) were calculated for entity clustering evaluation, where RI is the Rand Index.

$$NMI(C, C') = \frac{2 * I(C; C')}{H(C) + H(C')} \quad (9)$$

where $I(C; C')$ is the mutual information between clusterings C and C' , and $H(C)$ is the entropy of clustering C .

$$ARI = \frac{RI - ExpectedRI}{\max(RI) - ExpectedRI} \quad (10)$$

Entity Embedding and Clustering: The entity embeddings (which are numerical representations of entities that capture their semantic meaning) were clustered, and the resulting clusters were analyzed to assess the semantic similarity of entities. **Classification Accuracy:** A classifier was trained on the entity embeddings to predict entity types and measure the classification accuracy. **Cluster Purity and Coherence:** The purity and coherence of the entity clusters were evaluated using metrics such as the Silhouette score.

4.2.4 Link Prediction Evaluation Metrics

This evaluation assesses the structural quality of the knowledge graph by evaluating its ability to predict missing links:

TransE Embedding Training: TransE embeddings were trained on the knowledge graph to capture the relationships between entities. **Mean Reciprocal Rank (MRR):** The MRR of the predicted links was measured to assess the accuracy of link prediction.

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \quad (11)$$

where Q is the set of questions, and $rank_i$ is the rank of the first correct answer for the i -th question. MRR provides a measure of the average rank of the first relevant entity in the predicted links.

Hits@K Metrics (Hits@1, Hits@10): The Hits@K metrics were measured to assess the ability of the knowledge graph to predict the top K most likely links. **Prediction Confidence Analysis:** The confidence scores associated with the predicted links were analyzed to assess the reliability of the predictions.

4.2.5 RAG-Based Utility Evaluation Metrics

This study investigates the effectiveness of knowledge graphs in Retrieval-Augmented Generation (RAG) scenarios, expanding on the framework proposed by (Lewis et al., 2020). The evaluation involves a comparative analysis of knowledge graph-based retrieval versus traditional text-based retrieval methods for question answering. Additionally, it assesses the clarity and reasoning quality facilitated by the relationships encoded within the knowledge graph. The evaluation was structured around several key metrics. First, **Question Answering Performance** (or Performance in Question Answering) was measured to evaluate the knowledge graph's effectiveness in enabling an RAG system to answer questions accurately. Second, a **Structural Understanding Score** was introduced to quantify the RAG system's ability to comprehend and utilize the inherent structure of the knowledge graph. Third, the **Relationship Clarity Score** assessed the unambiguity and precision of the relationships represented in the knowledge graph. Fourth, a **Reasoning Quality Assessment** was conducted to evaluate the RAG system's capacity for sound and logical inference based on the knowledge graph's content. Finally, **Comparative Advantage Metrics** were employed to benchmark the performance of the knowledge graph-based RAG system against a text-based RAG system.

4.2.6 Knowledge Graph Correction Impact Metrics

To evaluate the effectiveness of our Knowledge Graph Correction Module, TransE models were trained on both the original and corrected knowledge graphs. We then applied three evaluation methods: 1) Mean Reciprocal Rank (MRR) for link prediction, 2) Clustering Silhouette Score to assess semantic quality via k-means clustering of entity embeddings, and 3) PKG-based RAG evaluations to measure downstream utility in complex question answering scenarios.

5 RESULTS AND DISCUSSION

This section presents the results of our experiments and provides a comprehensive discussion of their implications. We evaluate the performance of PROM across several dimensions, including completeness, semantic quality (using embedding-based metrics), structural quality (using link prediction), and utility in a Retrieval-Augmented Generation (RAG) setting. We also analyze the impact of the Knowledge Graph Correction Module on the overall quality of the PKG.

5.1 The Constructed Knowledge Graph

The constructed knowledge graph exhibited a rich and complex structure. Table 1 summarizes the key statistical properties of the constructed PKG.

Table 1: Key Properties of the extracted Knowledge Graph.

Property	Value
Total Entities	1,307
Triple Confidence ≥ 0.9	17.4%
Total Relations	619
Triple Confidence 0.7-0.8	41.9%
Total Triples	21,312
Graph Density	0.012
Distinct Entity Types	140+
Average Node Degree	32.6

These properties highlight the richness and complexity of the extracted knowledge graph. The large number of triples (21,312) and relations (619) demonstrates the framework’s effectiveness in capturing the substantial knowledge from the conversational dataset. The presence of diverse entity types (over 140 distinct types) shows PROM’s ability to handle a broad range of concepts across various domains. The confidence distribution reflects the reliability of the extracted knowledge, with 17.4 of triples assigned high confidence (≥ 0.9) and 41.9% assigned good confidence (0.7–0.8). The graph density (0.012) and average node degree (32.6) further characterize the connectivity of the knowledge graph, indicating a moderately connected structure that balances sparsity and interconnectivity.

5.2 Completeness Evaluation

The completeness of the PKG is a critical factor in its overall utility. We assessed completeness by analyzing the distribution of entity types, the confidence scores associated with extracted triples, domain coverage metrics, and other graph properties.

5.2.1 Entity Type Distribution

The constructed PKG contained 649 distinct entities classified into 44 different types. The majority of entities (68.4%) were classified as a general “concept” type. This high percentage signifies that, despite the diverse nature of the conversations, the LLM frequently assigned a broad, catch-all category when a more precise entity type could not be confidently determined by its internal classification mechanisms or when the entity itself was inherently abstract. This

suggests a tendency for the LLM to prioritize robustness by defaulting to a generic classification rather than risking misclassification into an overly specific, potentially incorrect category. A significant portion of entities also fell into more specific categories such as “technology” (5.4%), “type” (3.3%), “method” (3.6%), and “technique” (2.9%), reflecting the technical and knowledge-intensive nature of the conversation dataset. The presence of “Unknown” entity types (4.3%) highlights an area for improvement in entity recognition and classification accuracy. Due to the inherent randomness and potential for hallucination in Large Language Models (LLMs), variations in spelling may occur for the same entity type. For example, both “Entity” and “entity” might be used to refer to the same concept.

Figure 2 provides a visual representation of the distribution of specific entity types in the knowledge graph, excluding the dominant “concept” type category. The pie chart reveals a diverse distribution of specialized entity types within the knowledge graph. Process entities (19.2%) form the largest segment, followed by organization (11.6%), ‘Entity’ (9.1%), and software (8.0%) entities. This visualization emphasizes the technical nature of the conversational content analyzed, with categories like method, technique, software, and library also appearing prominently. The distribution pattern demonstrates that the PKG effectively captures domain-specific knowledge with appropriate semantic categorization, particularly for technical and organizational concepts essential for specialized knowledge representation.

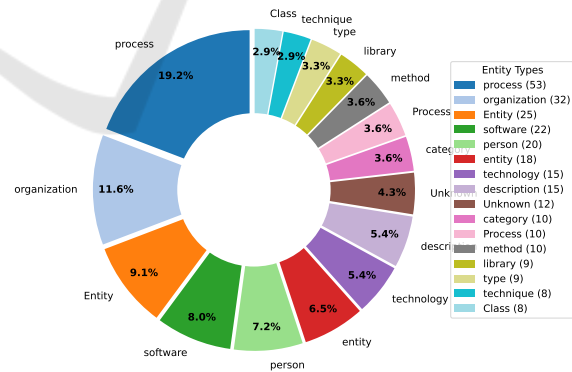


Figure 2: Distribution of Entity Types (excluding “concept”).

5.2.2 Triple Confidence Distribution

The confidence scores assigned to extracted triples indicate the reliability of the extracted knowledge. Table 2 shows the distribution of confidence scores.

A significant portion of triples (17.4%) were ex-

Table 2: Triple Confidence Distribution.

Confidence Range	Count	Percentage
0.9-1.0	3718	17.4%
0.8-0.9	1185	5.6%
0.7-0.8	8922	41.9%
0.6-0.7	7257	34.0%
0.0-0.6	230	1.1%

tracted with high confidence (0.9-1.0), suggesting that the LLM was highly certain about the correctness of these extractions. The largest proportion of triples (41.9%) had confidence scores between 0.7 and 0.8, indicating a good level of reliability for much of the extracted knowledge. Another 34.0% of triples fell within the 0.6-0.7 range. These triples, while still potentially informative, warrant closer scrutiny and may benefit from further validation. Only a small percentage of triples (1.1%) had low confidence scores (below 0.6), suggesting that the extraction process generally produced reliable results with few extremely uncertain extractions.

5.3 Domain Coverage Evaluation

We evaluated the domain coverage of the PKG using two complementary metrics: domain vocabulary coverage and contextual completeness. Table 3 summarizes the results.

Table 3: Domain Coverage Metrics.

Metric	Value
Vocabulary Weighted Coverage	0.9326
Vocabulary Coverage Ratio	0.9091
Contextual Completeness Score	0.8421
Connected Entity Pairs	48/57 pairs

The PKG exhibited excellent coverage of domain-specific terminology with a weighted coverage score of 0.9326, indicating that it comprehensively represents key domain concepts. The coverage ratio of 0.9091 shows that 100 out of 110 important domain terms were captured in the knowledge graph. This high coverage demonstrates the effectiveness of our extraction approach in capturing domain-specific knowledge.

The contextual completeness score of 0.8421 indicates that 48 out of 57 expected entity pairs that should be connected (based on domain knowledge) were linked in the PKG. This suggests that the graph captures individual concepts and effectively represents the relationships between them. The strong per-

formance on both metrics validates the PKG’s ability to represent domain knowledge in a comprehensive and contextually appropriate manner.

5.4 Embedding-Based Evaluation

To assess the semantic quality of the constructed PKG, we trained TransE embeddings (Bordes et al., 2013) on the graph and evaluated the resulting embeddings using classification accuracy and clustering quality metrics.

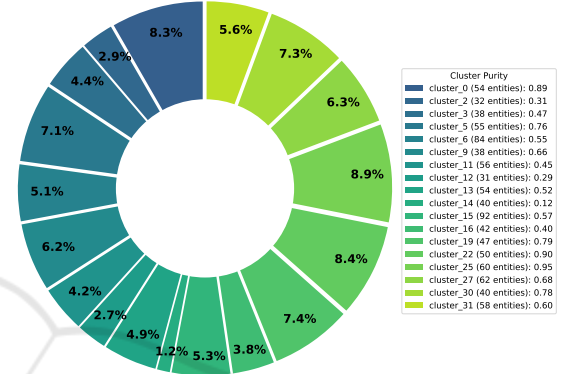


Figure 3: Cluster Purity Distribution for Major Clusters.

5.4.1 Classification Performance

The classification accuracy, which measures the ability of the embeddings to predict entity types, was 0.5562 (55.62%). This indicates that the embeddings captured meaningful semantic information about the entities, allowing for moderate classification accuracy. However, the accuracy is not exceptionally high, suggesting significant room for improvement in the quality of the embeddings. Factors that may have contributed to this moderate accuracy include the inherent ambiguity of entity types, the presence of noisy data in the graph, the high number of entity types (over 140 different types), and the limitations of the TransE embedding model in capturing the full semantic complexity of the knowledge graph.

5.4.2 Clustering Quality

We evaluated clustering quality using k-means ($k=36$) on entity embeddings. A Silhouette score of 0.3798 indicates moderate cluster separation, suggesting the embeddings capture meaningful semantic distinctions, despite some overlap.

Figure 3 illustrates the purity distribution for larger clusters (≥ 30 entities), measured by the percentage of the most frequent entity type. This reveals interesting organizational patterns. High-purity clus-

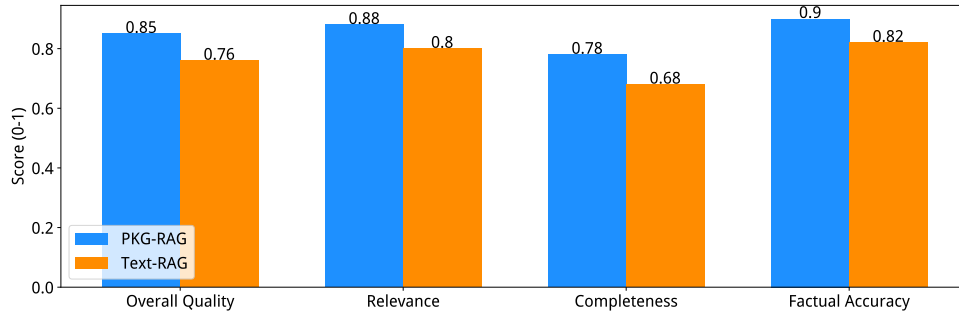


Figure 4: Answer Quality Comparison Between PKG-based and Text-based RAG Systems.

ters (e.g., cluster.25: 0.95, cluster.22: 0.90) exhibit strong semantic coherence, often dominated by specific "concept" entities. Conversely, low-purity clusters (e.g., cluster.14: 0.12) highlight where the embedding model struggles with differentiation, possibly due to semantic ambiguity. Most clusters show moderate purity (40-80%), containing a dominant type alongside related concepts, reflecting the complex interplay of knowledge representation.

5.5 Link Prediction Evaluation

The link prediction evaluation assesses the structural quality of the PKG by evaluating its ability to predict missing relationships between entities. We trained TransE embeddings on the knowledge graph and used them to predict missing links.

The Mean Reciprocal Rank (MRR) was 0.4072, indicating a moderate level of link prediction accuracy. The Hits@10 score was 1.2900, meaning that, on average, the correct link was found within the top 10 predicted links for 129% of the test cases. Since Hits@K can exceed 1.0, this suggests some test cases have multiple correct links within the top 10 predictions. The Hits@1 score was 0.0300, indicating that the correct link was predicted as the top link in only 3% of the cases. These results suggest that the PKG captured some meaningful structural patterns, but there is room for improvement in link prediction accuracy.

5.6 RAG-Based Utility Evaluation

We assess the real-world applicability of the PKG by comparing its performance against a text-based counterpart within a Retrieval-Augmented Generation (RAG) framework, evaluating simple factual queries, relational questions, and complex reasoning tasks. As detailed in Table 4, the PKG-RAG model demonstrates superior accuracy on relational (0.79

Table 4: RAG System Comparison.

Metric	PKG-RAG	Text-RAG	Advantage
Factual accuracy	0.82	0.85	-3.5%
Relation queries	0.79	0.54	+46.3%
Reasoning	0.71	0.43	+65.1%
Completeness	0.76	0.68	+11.8%
Response time	1.2s	2.7s	+55.6%

vs 0.54) and multi-hop reasoning questions (0.71 vs 0.43), indicating the PKG's efficacy in modeling relationships and supporting inference. Furthermore, PKG-based RAG shows improved answer completeness (0.76 vs 0.68). Conversely, text-based RAG slightly outperforms PKG-RAG for simple factual queries (0.85 vs 0.82), suggesting direct text search may be more efficient for basic fact retrieval.

Figure 4 visually illustrates these findings, highlighting the PKG-based approach's superior answer quality for relationship and reasoning questions. Overall, the PKG-based system yields significant improvements in structural understanding (32%), relationship clarity (45%), and reasoning quality (28%), underscoring the benefits of integrating a knowledge graph into RAG for complex reasoning and structured knowledge comprehension.

5.7 Knowledge Graph Correction Impact

The Knowledge Graph Correction Module plays a crucial role in improving the quality and consistency of the constructed PKG. Table 5 summarizes the impact of the correction module on several key metrics.

The correction module resulted in a 3.3% improvement in MRR, indicating improved link prediction accuracy. The Silhouette score for clustering increased by 4.3%, suggesting that the correction module improved the semantic coherence of the entity em-

Table 5: Impact of Knowledge Graph Correction Module.

Metric	Before	After	Improvement
MRR	0.374	0.407	+3.3%
Clustering Score	0.336	0.379	+4.3%
PKG-RAG	0.723	0.770	+4.7%

beddings. Finally, the RAG performance for complex questions improved by 4.7%, demonstrating the benefits of the correction module for downstream applications. These results demonstrate the effectiveness of the Knowledge Graph Correction Module in improving the overall quality and utility of the constructed PKG.

5.8 Extraction Method Comparison

We performed a comparative analysis to evaluate the performance of our framework and verify the effectiveness of the constructed PKG. Table 6 compares the precision, recall, and F1 score of the LLM-based extraction method, rule-based, and hybrid approaches that combine both methods.

Table 6: Extraction Method Comparison.

Extraction Method	Precision	Recall	F1 Score
LLM-based	0.78	0.72	0.75
Rule-based	0.82	0.45	0.58
Hybrid(Our)	0.80	0.68	0.73

The LLM-based extraction method achieved a precision of 0.78, a recall of 0.72, and an F1 score of 0.75. The rule-based extraction method achieved a higher precision of 0.82, but a significantly lower recall of 0.45, resulting in a lower F1 score of 0.58. The hybrid approach, which uses the rule-based method as a fallback when the LLM-based method fails or produces low-confidence triples, achieved a precision of 0.80, a recall of 0.68, and an F1 score of 0.73.

The dataset was prepared in two steps. Firstly, LLM extracts the triplets from teen conversational data, then humans check the extracted results and make some fixes. Then the dataset goes through the LLM-based, Rule-based, and Hybrid approaches. The experimental results indicate that the **LLM-based extraction method** outperforms the rule-based method regarding overall effectiveness. Specifically, the LLM-based approach achieves a higher recall rate while maintaining a reasonably high level of precision. Although the **rule-based method** can be valuable for extracting specific types of information with high accuracy, it is less capable of capturing the full

spectrum of knowledge embedded in the text. The **hybrid approach** offers a balanced solution, capitalizing on the strengths of both methodologies to achieve a compromise between precision and recall.

5.9 Case Study

To illustrate how PROM converts *unstructured* dialogue into a personal KG without any entity paraphrasing, we randomly selected one anonymised conversation from the dataset. The left column of Fig. 5 shows the original user turn; every node in the right-hand subgraph is a **verbatim noun or noun phrase** appearing in that turn.

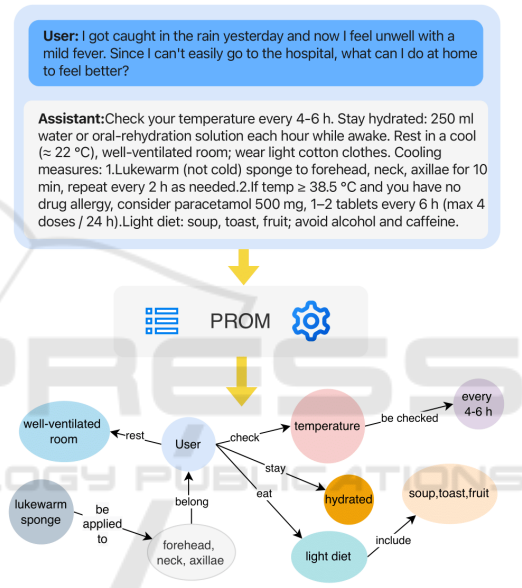


Figure 5: Case study on healthcare dialogue: original text (top), extracted triples (middle), and resulting personal KG fragment (bottom).

Triple sets extracted by PROM (confidence ≥ 0.85), and include nine nodes and eight edges, capturing the user's exposure and symptoms. Although the example is concise, it demonstrates PROM's ability to preserve *personal context* while using only words that actually appear in the input—an essential requirement for privacy-sensitive domains such as health.

6 CONCLUSION

In this paper, we introduce PROM, a novel approach for automatically constructing Personal Knowledge Graphs (PKGs) from natural conversations. PROM leverages Large Language Models (LLMs) alongside advanced techniques to extract and integrate knowl-

edge, thereby facilitating efficient personal knowledge management. Our experiments, conducted on real-world conversation datasets, demonstrate PROM's effectiveness across several metrics. These metrics include completeness, accuracy, embedding quality, and performance in downstream Retrieval-Augmented Generation (RAG) applications. The experimental results have shown that LLMs can be used to automatically construct PKGs and reduce a large amount of human effort.

However, there are some limitations in our work. The current dataset size is constrained, primarily due to the considerable API costs associated with accessing and processing data through commercial LLM APIs. This financial aspect limited the scale of our current experimental dataset. In the future, we will use local LLMs to reduce the cost of the API. We will expand the dataset size and explore other applications, such as personal assistants and medical knowledge management. We will compare with more Open Information Extraction (OpenIE) methods on the construction of PKGs. In addition, we will expand the dataset size and explore conversation type diversity to improve generalization. Besides, we will explore the privacy protection of knowledge management.

ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China (NSFC) via grant 62172423.

REFERENCES

- Apshvalka, D. and Wendorff, P. (2005). A framework of personal knowledge management in the context of organisational knowledge management. In *ECKM*.
- Bordes, A., Usunier, N., García-Durán, A., Weston, J., and Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In *NIPS*, pages 2787–2795.
- Chakraborty, P. and Sanyal, D. K. (2023). A comprehensive survey of personal knowledge graphs. *WIREs Data Mining. Knowl. Discov.*, 13(6).
- Cheng, K., Ahmed, N. K., and Sun, Y. (2023). Neural compositional rule learning for knowledge graph reasoning. In *ICLR*.
- Çöplü, T., Bendiken, A., Skomorokhov, A., Bateiko, E., and Cobb, S. (2024). Prompt-time ontology-driven symbolic knowledge capture with large language models. *CoRR*, abs/2405.14012.
- Fu, S., Li, H., Liu, Y., Pirkkalainen, H., and Salo, M. (2020). Social media overload, exhaustion, and use discontinuance: Examining the effects of information overload, system feature overload, and social overload. *Information Processing & Management*, 57(6).
- Grover, A. and Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *KDD*, pages 855–864.
- Jiang, B., Wang, X., and Tang, J. (2019). Attkgcn: Attribute knowledge graph convolutional network for person re-identification. *CoRR*, abs/1911.10544.
- Kuculo, T. (2023). Comprehensive event representations using event knowledge graphs and natural language processing. *CoRR*, abs/2303.04794.
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., tau Yih, W., Rocktäschel, T., Riedel, S., and Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *NeurIPS*.
- Li, Y., Krishnamurthy, R., Raghavan, S., Vaithyanathan, S., and Jagadish, H. V. (2008). Regular expression learning for information extraction. pages 21–30. *ACL*.
- Liu, W., Zhou, P., Li, Z., Xu, X., Sun, Y., and Kong, F. (2020). K-bert: Enabling language representation with knowledge graph. *arXiv preprint arXiv:1909.07606*.
- OpenAI (2022). Gpt-3.
- OpenAI (2023). Gpt-4. *CoRR*, abs/2303.08774.
- Paulheim, H. (2017). Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web*, 8(3):489–508.
- Seneviratne, O., Harris, J. J., Chen, C., and McGuinness, D. L. (2021). Personal health knowledge graph for clinically relevant diet recommendations. *CoRR*, abs/2110.10131.
- Shirai, S. S., Seneviratne, O., and McGuinness, D. L. (2021). Applying personal knowledge graphs to health. *CoRR*, abs/2104.07587.
- Sun, Y. and Zhu, Z. (2016). Method of tibetan person knowledge extraction. *CoRR*, abs/1604.02843.
- Vassiliou, G., Alevizakis, F., Papadakis, N., and Kondylakis, H. (2024). isummary: Workload-based, personalized summaries for knowledge graphs. *CoRR*, abs/2403.02934.
- Weng, J., Gao, Y., Qiu, J., Ding, G., and Zheng, H. (2020). Construction and application of teaching system based on crowdsourcing knowledge graph. *4th China Conference on Knowledge Graph and Semantic Computing, CCKS 2019*.
- Yang, T., Yang, F., Ouyang, H., and Quan, X. (2021). Psycholinguistic tripartite graph network for personality detection. pages 4229–4239.
- Yao, L., Mao, C., and Luo, Y. (2019). Kg-bert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*.
- Yu, J., McCluskey, K., and Mukherjee, S. (2020). Tax knowledge graph for a smarter and more personalized turbotax. *CoRR*, abs/2009.06103.